

# Is the Measurement Based Quantum Computing Search Algorithm Really Grover's Algorithm?

A. Matthew Smith<sup>1</sup>, P. M. Alsing<sup>1</sup>, J. R. McDonald<sup>1</sup> and D. B. Uskov<sup>2</sup>

<sup>1</sup>*Air Force Research Laboratory, Information Directorate, Rome, New York 13440, USA*

<sup>2</sup>*Dept. of Mathematics and Natural Science, Brescia University, Owensboro, Kentucky 42301, USA*

We question whether the measurement based quantum computing algorithm is in fact Grover's algorithm or simply a similar oracular search method. The two algorithms share several qualitative features especially in the case of the trivial 4 element search, which is the largest size photonic search algorithm that has been experimentally implemented to date. This has led some to refer to both substantiations as Grover's algorithm. We compare multiple features of the two algorithms including the behavior of the oracle tags and the entanglement dynamics, both qualitatively and quantitatively. We find significant and fundamental differences in the operation of the two algorithms, particularly in cases involving searches on more than four elements.

PACS numbers: 03.67.Ac,03.67.Lx

## I. INTRODUCTION

L. Grover first showed that a speed up over classical database searches was possible using quantum mechanics. His search algorithm, now known as Grover's algorithm [1], along with Shor's factoring algorithm [2] helped to spur the interest in all manners of quantum computing. Multiple implementations of universal quantum computing have been found and tested, such as Linear Optical Quantum Computing LOQC [3], Adiabatic Quantum Computing (AQC) [4] and Measurement Based Quantum Computing (MBQC) or equivalently cluster state quantum computing (CSQC) [5, 6] just to name a few. Grover's algorithm has been shown by C. Zalka [7] to be the optimal means of searching an unsorted list via amplitude amplification and, with some minor modifications, has been shown to have near perfect fidelity in the limit of large system size  $N$  [8]. We will compare and contrast the MBQC search with Grover's algorithm in the photonic circuit model. In Section II we describe the initial assumption that led to this line of inquiry. In Section III we describe in detail the MBQC implementation and our numerical framework. Section IV explicitly defines the two algorithms. Section V describes one of the main differences between the algorithms, namely the tagging operation in the MBQC search. Section VI deals with amplitude amplification and compares the speed of the two algorithms. Section VII discusses the differences in the entanglement dynamics. Section VIII asks several unresolved questions about the MBQC search. Section IX summarizes our results and makes concluding remarks.

## II. ASSUMPTION THAT CIRCUIT MODEL AND MEASUREMENT BASED MODEL GROVER SEARCH ALGORITHMS ARE THE SAME.

In almost every implementation of quantum computing one of the first algorithms that people have developed and tested is an unstructured quantum search algorithm. In the photonic circuit model of quantum computing [9] this algorithm is structurally very similar to that first proposed by Grover and as such is called Grover's algorithm. Since the development of MBQC, and particularly in the photonic implementation of MBQC, also called the cluster state model, parallels have been drawn between the photonic circuit model and MBQC model. This includes comparing the circuit model of Grover's algorithm to a cluster state model of a 4 element quantum search [10]. Given the difficulty of creating large cluster states this four element search is the largest that has been performed to date. The,  $2^n = 2^2 = 4$  element search requires  $n^2 = 2^2 = 4$  qubits arranged in a square or "box" cluster state [10]. The next larger  $2^3 = 8$  element search requires a significantly more difficult to produce 3x3 square cluster state and so on.

Using the 4 qubit box state, the MBQC search appears to be very similar to that of Grover. The input state is similar, the desired output state is similar, the number of iterations is the same and the oracle's tagging operation appears similar. This has led many people to call the MBQC quantum search, Grover's algorithm. However there is often a small caveat that is overlooked. As Zeilinger *et al.* wrote "Remarkably, the inversion-about-the-mean process is 'hard-wired' into the structure of the cluster state and is automatically implemented" [10]. We found this automatic implementation of an inversion about a mean, or equivalently amplitude amplification (when combined with tagging), to be an imprecise claim and as such have investigated the measurement based quantum search algorithm numerically for larger systems. We have found that a significant number of non-trivial differences exist between the standard description of Grover's algorithm and its implementation in the circuit model and the MBQC search algorithm for

larger systems. This leads us to question whether the MBQC search algorithm is some variant of Grover's algorithm, or rather a different search algorithm altogether.

### III. THE MEASUREMENT BASED QUANTUM COMPUTING AND THE NUMERICAL FORMALISM

Measurement based quantum computing (MBQC) operates by first generating highly entangled multi-qubit states called cluster states. Computations are then performed by a time ordered sequence of measurements on individual qubits in this entangled resource. Some cluster states, such as the 2 dimensional square cluster state (see Fig. (1)), have been proven to be able to perform universal computation assuming the initial cluster state is large enough.

To begin the construction of the cluster state, we first form the Kronecker product of  $N$  photons each initialized to the  $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$  state.

$$|\psi_{init}\rangle \equiv \prod_{i=1}^N |+_i\rangle = |+_1\rangle \otimes |+_2\rangle \otimes \dots |+_N\rangle. \quad (1)$$

However, to complete the construction of the cluster state we must create the entanglement resource between the qubits by effectively performing the CZ operation between nearest neighbors. This entangling operation between two qubits can equivalently come from other sources such as spontaneous parametric down conversion (SPDC) [11]. The four CZ operations for the box cluster state (see Fig.(1a)) are shown in Eq.(2),

$$|\Psi_0\rangle = |\Psi_{box}\rangle = \prod_{i,j=nn} CZ_{i,j} |\psi_{init}\rangle = CZ_{1,2} CZ_{1,3} CZ_{2,4} CZ_{3,4} |\psi\rangle, \quad (2)$$

where  $CZ_{i,j}$  is a  $4 \times 4$  matrix with entries  $(-1)^{b_i b_j}$  for the binary computational basis entries  $b_i, b_j \in \{0, 1\}$ . Here we note that  $i, j$  are the labels of the control and target qubits respectively in the CZ operation,  $nn$  is a list of all nearest neighbors and the that order of the CZ operations is irrelevant. Two of the cluster states of interest in the remainder of this paper are then shown in Fig. (1); the 4 qubit "box" state  $|\Psi_{box}\rangle$  and its next logical extension, the 9 qubit square cluster state  $|\Psi_{C9}\rangle$ . The  $CZ_{i,j}$  operations create the  $n \times n$  square cluster state the MBQC search requires with  $N = n^2$  total qubits.

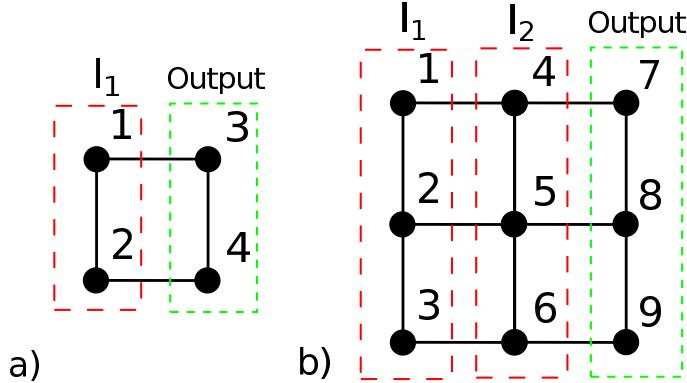


FIG. 1: **Square Cluster-states.** Two cluster-states able to perform an unsorted search for a) the ( $n = 2$ )  $2^2 = 4$  element search on  $|\Psi_{box}\rangle$  and b) the ( $n = 3$ )  $3^2 = 8$  element search on  $|\Psi_{C9}\rangle$ . Each circle is a single qubit initialized in the  $|+\rangle$  state. The solid lines connecting the circles indicate the action of a CZ gate between these qubits, i.e. entanglement between the qubits. Logical qubits are arranged in rows of the grid, initially: (1, 3) and (2, 4) for  $|\Psi_{box}\rangle$ , and (1, 4, 7), (2, 5, 8), and (3, 6, 9) for  $|\Psi_{C9}\rangle$ . a) shows the trivial MBQC measurements using one application of an iterator  $I_1$  (large red dashes acting on columns of qubits) on the 4 qubit box state. b) Shows the two iterations for the 8 element search  $I_1, I_2$ . All of the measurements in an iterator  $I_j$  can be implemented simultaneously. The outcomes of each  $I_j$  determine the basis of the next iteration  $I_{j+1}$ . (Equivalently  $I_j$  determines the Pauli correction factors that can be applied latter in post-processing). The output state (small green dashes) holds the logical state at the end of the algorithm. The effective logical qubits after the first iteration  $I_1$  are (3) and (4) (the output states) for  $|\Psi_{box}\rangle$ , and (4, 7), (5, 8), and (6, 9) for  $|\Psi_{C9}\rangle$ .

An algorithm then proceeds as a series of single-qubit measurements. The bases for the measurements performed at the  $(k + 1)$ th iteration are determined by the outcomes of the previous measurements performed at iteration  $k$ . This

can be considered as an ordered series of Kronecker products of single qubit rotations. Equivalently we may rotate the measurement basis. Such a Kronecker product of rotations produces a unitary matrix. For the MBQC search algorithm we need only the set of rotations of  $s = \{0, \pi\}$  [10]. We are measuring in the basis  $|\pm_s\rangle = (|0\rangle \pm e^{is}|1\rangle)/\sqrt{2}$ , i.e. either the standard  $|\pm\rangle$  basis ( $s = 0$ ) or the  $|\mp\rangle$  basis ( $s = \pi$ ).

The MBQC search proceeds by implementing measurements on the  $(n - 1)$  vertical columns of qubits, each of size  $n$  on the cluster state. It proceeds sequentially from left to right, leaving the last (i.e. rightmost) column of  $n$  qubits as the output state of the computation, see Fig. (1). Operationally, each measurement  $B(\alpha)$  is implemented by the product of a rotation matrix  $R(\alpha)$  times the Hadamard matrix  $H$ ,

$$B(\alpha) = R(\alpha)H, \quad R(\alpha) = \begin{pmatrix} \cos \alpha/2 & i \sin \alpha/2 \\ i \sin \alpha/2 & \cos \alpha/2 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (3)$$

followed by a projection  $P$  onto  $|\pm_s\rangle$ . Note that the rightmost  $H$  in  $B(\alpha)$  rotates the computational basis states  $\{|0\rangle, |1\rangle\}$  (eigenstates of  $Z$ ) into the states  $\{|+\rangle, |-\rangle\}$  (eigenstates of  $X$ ) respectively, while  $R(\alpha)$  effects the rotations  $R(\alpha)|\pm\rangle = e^{\pm i\alpha/2}|\pm\rangle$ . In this formalism the unitary matrices that implement the measurement based search algorithm takes the following forms for the two search sizes of interest shown in Fig.(1):

$$U_{box} = B_1(\alpha) \otimes B_2(\beta) \otimes H_3 \otimes H_4, \quad (4a)$$

$$U_{C9} = B_1(\alpha) \otimes B_2(\beta) \otimes B_3(\gamma) \otimes B_4(a) \otimes B_5(b) \otimes B_6(c) \otimes H_7 \otimes H_8 \otimes H_9. \quad (4b)$$

The output state of the calculation, prior to the the projection operations, is then,

$$|\Psi_{out}\rangle = U|\Psi_{in}\rangle. \quad (5)$$

The output state  $|\Psi_{out}\rangle$  is a vector of size  $2^N$ . The output state can divided into  $2^{N-n}$  sections of size  $2^n$ , each of which correspond to a set of specific measurements outcomes on the  $n - 1$  columns of  $n$  qubits. In the first section of  $2^n$  elements of the state  $|\Psi_{out}\rangle$ , each of the prior  $n(n-1)$  qubits have been projected “correctly” into the state  $|+_s\rangle$  (i.e  $|+\rangle$  for  $s = 0$ , and  $|-\rangle$  for  $s = \pi$ ) for their given value of  $s$  (i.e.  $|\Psi_{out}\rangle = |+_s\rangle_1 \otimes \dots \otimes |+_s\rangle_{N-n} \otimes |\psi_{out}\rangle_{(N-n)+1, \dots, N}$  where  $|\psi_{out}\rangle_{(N-n)+1, \dots, N}$  is the output state on the last column of  $n$  qubits) yielding the output for the MBQC search algorithm. Each of the remaining segments, each of size  $2^n$ , correspond to “incorrect results” in which one or more of the prior  $n(n - 1)$  qubits has been projected into the  $|-_s\rangle$  for their given value of  $s$ . However, like any MBQC algorithm these errors are known, since we know the outcomes of the measurements, and they can be corrected during or after the calculation with polynomial resource overhead. For full details see [12] or other texts on MBQC [13]. The essential point is that these “incorrect outputs” are calculable and can be trivially mapped back into the correct output based on the measurement outcomes for any size system. Therefore, we can assume without loss of generality that the single qubits measurements all give the “correct” output. This eliminates the need to deal with trivial feed forward corrections at the end of the algorithm. We can apply the projection matrix  $P = |+_s\rangle\langle+_s|$  to each measurement (See Appendix A for details). The measurement operations for the 4 qubit box Eq.(4a) are then,

$$U_{box} = PB_1(\alpha) \otimes PB_2(\beta) \otimes H_3 \otimes H_4, \quad P = |+_s\rangle\langle+_s| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \quad (6)$$

This projective operation populates the unitary matrix  $U_{box}$  with the non-zero values in the first  $2^n$  rows and zeros in the rest of the matrix. This is convenient for calculations involving large cluster states for which using dense matrices of size  $2^{n^2} \times 2^{n^2}$  is computational expensive. We can simply assume the measurements in the MBQC scheme proceed “correctly” and truncate the matrix to size  $2^n \times 2^{n^2}$ . Note that  $U$  is unitary in the absence of the projection matrix  $P$ . In the remainder of the work the projective measurements  $P$  are implied on the first  $n \times (n - 1)$  qubits.

We sample the  $2^n$  outputs by simply rotating the  $n$  output qubits of the last column of the grid to the  $|\pm\rangle$  basis. This is equivalent to preforming a simple Hadamard on each output qubit and measuring. Note that different possible measurement patterns on the cluster state are equivalent (right to left, left to right, top to bottom and bottom to top); one simply has different locations for the output state.

#### IV. GROVER'S ALGORITHM VS. THE MBQC SEARCH ALGORITHM

In this section we examine the details of Grover's algorithm and compare it to our implementation of the MBQC search algorithm.

### A. Grover's algorithm

Consider an unstructured set of  $N_G = 2^n$  elements  $X = \{1, \dots, N_G\} \in \{0, 1\}^{\otimes n}$  for which we desire to find a specific element  $x_0$ . For  $x \in X$  let  $f(x) : \{0, 1\}^{\otimes n} \rightarrow \{0, 1\}$  be an externally supplied function (the *oracle*) defined such that  $f(x_0) = 1$  and  $f(x \neq x_0) = 0$ . Grover defined his algorithm with the following steps [1].

- 1) initialize the system of size  $N_G$  to the uniform amplitude, unbiased state  $|\psi_{init}\rangle = \frac{1}{\sqrt{N_G}} \sum_x |x\rangle \equiv (1, 1, \dots, 1)/\sqrt{N_G}$  (in the computational basis of states  $\{|0\rangle, |1\rangle\}^{\otimes n}$ ).
- 2) Repeat the following operations  $O(\sqrt{N_G})$  times:
  - a) Let  $|x\rangle$  be any computational basis state:
    - if  $f(x_0) = 1$  rotate the state by  $\pi$ ,
    - if  $f(x) = 0$  leave the state unaltered.
  - b) Apply the diffusion transform  $D$  such that:
 
$$D_{ij} = 2/N_G \text{ if } i \neq j \text{ & } D_{ii} = -1 + 2/N_G.$$
- 3) Sample the state by projective measurement.

It is common to refer to step “2a” as a tagging operation in which the phase of the desired state  $|x_0\rangle$  (or set of states, in general) is given a  $\pi$  phase flip. Step “2b” is the so called inversion about the mean, which is essentially a rotation that performs an averaging of the state amplitudes. Both “a” and “b” can be written as unitary operators, which we will denote  $U_f$  and  $U_{inv}$ . The iterator of Grover's algorithm is then simply the matrix product  $U_G \equiv U_{inv}U_f$ . Grover's algorithm begins from a uniform input state of size  $N_G = 2^n$  and performs a sequence of repeated tag and inversion operations  $U_G^k$  until the desired number of steps  $k \approx \mathcal{O}(\sqrt{N_G})$  have been performed. The state is then measured and the desired result or results will have an overwhelming likelihood of being measured.

As is well known, the tagging operator  $U_f$  requires knowledge of the solution element sought in order to be explicitly constructed [14]. This is why the oracle  $U_f$  is part of the “input” to the Grover search algorithm, and it has to be supplied *externally* (via an oracle) to the user performing the search. Here,  $U_f$  is the unitary implementation of the classical function  $f$ . In order to be unitary (and hence reversible),  $U_f$  acts on the product  $n$ -qubit basis states  $|x\rangle$  and a single auxiliary qubit  $|y\rangle$  as  $U_f|x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x)\rangle$ , where  $\oplus$  denotes binary (mod 2) addition. For  $|y\rangle = |- \rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ , explicit computation reveals that  $U_f|x\rangle \otimes |- \rangle = (-1)^{f(x)}|x\rangle \otimes |- \rangle$ . This *phase kickback* operation is often abbreviated as  $U_f|x\rangle = (-1)^{f(x)}|x\rangle$ , though the use of the single auxiliary qubit is mandatory in order to ensure the unitarity of  $U_f$ . An explicit representation of tagging operator is given by [14]  $U_f = |x_0\rangle \langle x_0| \otimes X_2 + \sum_{x \neq x_0} |x\rangle \langle x| \otimes I_2$ , where  $X_2$  and  $I_2$  are the  $2 \times 2$  single qubit Pauli X bit-flip operator and identity matrix respectively, which act on the auxiliary qubit. This latter expression demonstrates that the oracle must know the sought after solution  $x_0$  in order to know where to place the single qubit bit-flip operator  $X_2$  in the block-diagonal matrix  $U_f$  (where all the remaining  $2 \times 2$  diagonal blocks for  $x \neq x_0$  are filled with  $I_2$ ). A variation of Grover's algorithm that circumvents this requirement to “know” the solution state  $x_0$  in order to construct the tagging operator  $U_f$  is given in [15] which searches on pairs (or more) of tensor product states  $|x\rangle \otimes |t\rangle$ , where say,  $x$  denotes a name, and  $t$  a telephone number. Here, searching on a supplied telephone number  $|t_0\rangle$  in order to find an associated unknown name  $|x_0\rangle$  requires the construction of the tagging operator  $I^x \otimes U_f^t$ , where  $I^x$  acts unbiasedly as the identity operator on the unknown subspace of names  $\{|x\rangle\}$ , and  $U_f^t$  acts in the subspace of telephone numbers.

### B. MBQC search algorithm

In this section we describe the standard MBQC search algorithm.

- 1) Initialize a set of  $N = n^2$  qubits into an  $n \times n$  square cluster state in which each qubit is in the plus state  $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$  and  $2^n$  is the maximum desired number of elements to be searched.
- 2) An oracle selects a set of measurements of size  $n \times (n - 1)$ , to be made in either  $s = 0$  or  $\pi$  basis such that  $|\pm_s\rangle = (|0\rangle \pm e^{is}|1\rangle)/\sqrt{2}$  and:
  - a)  $n - 1$  sets of measurements are applied sequentially, in which  $n$  qubits are measured simultaneously, moving across the cluster state in a uniform direction (i.e. left to right, or top to bottom), and such that the  $(k + 1)$ th measurement set depends on the outcomes (successful or unsuccessful) of the  $k$ -th measurement set, for all  $k = 1, \dots, n - 1$  iterations.
  - b) Sample the final state by rotating the remaining  $n$  qubits into the  $|\pm\rangle$  basis, followed by projective measurement.

The most apparent difference between the definition of Grover's search algorithm and the MBQC algorithm is in the form and function of the iterator. In the circuit model of Grover's algorithm the iterator is a product of two unitary operators which repeatedly implements the same (i.e. constant between iterations) tagging operation, followed by the

same inversion about the mean operation. In the MBQC search algorithm this repetitive implementation of fixed, constant iterations is not mandatory. The size  $n$  “iterator”, which consists of the  $n$  measurement bases chosen by the oracle, need not, and in general will not, be a constant between iterations.

## V. TAGGING BY THE ORACLE IN THE MBQC SEARCH ALGORITHM

### A. The MBQC search algorithm with general non-constant tagging operations

In Grover’s algorithm the tagging operation, which is preformed by the oracle, is a unitary transformation which simply flips the phase of a desired element. As a non-trivial, illustrative example, we consider searching for a unique element in a eight element list. This tagging operation forms half of the Grover iterator. The tagging is one to one, meaning there is a single unique tag for each and every element in the search space. In order for the inversion about the mean to take place the oracle must continue to apply the same tagging operation at each step, i.e. iteration, of the search. As stated above this is different from the MBQC search.

The MBQC search is unique only up to overall phases such as  $\pm$  signs and factors of  $i$ , and the final output states can be degenerate. For the  $(n = 3) 2^n = 8$  element search on the  $3 \times 3$  grid of qubits, the the final possible measurement outcomes on the last column of 3 qubits are 8 fold degenerate. That is, there are 8 tagging operations that all point to the same final state (ignoring overall phases). In the special case of the  $(n = 2) 2^n = 4$  element search implemented on a  $2 \times 2$  grid of qubits (the box cluster state), there are 4 measurements and 4 unique output states, implying that the tagging operations are one-to-one. We believe that the one-to-one mapping between the tagging operation and the final output state exhibited in the special case of  $n = 2$  has resulted in part of the confusion about the equality of Grover’s and the MBQC search algorithms. We have found that a one-to-one property associating a tagging operation to a unique output state does not scale with the size of the cluster state.

The MBQC search can be implemented successfully for any  $n$  with a constant (i.e. fixed) iterator. The  $(n = 3) 2^n = 8$  element search on 9 qubits arranged on a  $3 \times 3$  grid can be implemented with an iterator composed of three measurements  $a, b, c \in \{0, \pi\}$ . This iterator would be applied twice to the two columns of 3 qubits, i.e.  $(a, b, c ; a, b, c)$  on qubits  $(1, 2, 3 ; 4, 5, 6)$  respectively. However, this is not a necessary condition, and the MBQC search could instead be preformed with an iterator that varies between measurement iterations (see Appendix B for the 3 and 4 qubit examples).

Interestingly, we have found that if one ignores, or corrects for overall phase, then not all of the  $n \times (n - 1)$  measurements are strictly necessary. Note that this does not mean that some of the qubits are unnecessary, the size  $n^2$  of the qubit grid remains the same. As an example, we again use the  $(n = 3) 2^n = 8$  element search on 9 qubits on a  $3 \times 3$  grid. This search requires  $3 \times (3 - 1) = 6$  measurements chosen by the oracle with the remaining 3 output qubits rotated to the  $|\pm\rangle$  basis via an  $H^{\otimes 3}$  operation. However, we find that we can preform the full 8 element search, i.e. produce every possible output state (ignoring the overall phase of the output) by varying only  $n = 3$  measurements. Explicitly the sequence  $(0, 0, s_3 ; 0, s_5, s_6)$  on qubits  $(1, 2, 3 ; 4, 5, 6)$ , where 0 implies a measurement projection onto the standard plus state  $|+\rangle$ , i.e  $B(0)$ , and  $s_i$  implies a measurement that can be varied,  $s = 0$  or  $s = \pi$ , by the oracle based on the desired output (i.e. the oracle can change the  $s_i$ ’s to tag the desired output state, but doesn’t need to change any of the measurements involving  $B(0)$ ). It is important to note that not every combination of  $n = 3$  measurements gives the full space of  $2^n = 8$  outputs. As an example, the measurement pattern  $(0, 0, 0 ; s_4, s_5, s_6)$  only results in 4 distinct output states (up to overall phases), i.e. a 2 qubit search. The pattern of measurements  $(0, 0, s_3 ; 0, s_5, s_6)$  has a form similar to Eq.(4a),

$$U_{C9} = H_1 \otimes H_2 \otimes B_3(\gamma) \otimes H_4 \otimes B_5(b) \otimes B_6(c) \otimes H_7 \otimes H_8 \otimes H_9. \quad (7)$$

This leads to the question of precisely what operation the “tagging” unitary is performing on the state. Above we define the operation of an MBQC algorithm as a unitary matrix constructed by taking the Kronecker product of rotation operations on the  $n(n - 1)$  measured qubits, multiplied by the Hadamard matrices on the final  $n$  output qubits. This leads to the Kronecker product of a series of matrices chosen from the set of 0 or  $\pi$  rotations, or explicitly from Eq.(3),

$$B(0) = H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \& \quad B(\pi) = F = \frac{i}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}. \quad (8)$$

The MBQC search operation  $U$  is then the Kronecker product of  $H$ ’s and  $F$ ’s. For the  $(n = 3) 3 \times 3 = 9$  qubit cluster state  $U$  has the following form given in Eq.(9a) and Eq.(9b) for the  $n(n - 1) = 6$  measurements  $(\alpha, \beta, \gamma ; a, b, c)$ , with the output qubits 7, 8 & 9 rotated into the  $|\pm\rangle$  basis. Two specific examples of the set of  $2^{n(n-1)} = 64$  unitary

matrices are shown below that produce the output states  $|1\rangle$  and  $|4\rangle$ , or  $(1, 0, 0, 0, 0, 0, 0, 0)^T$  and  $(0, 0, 0, 1, 0, 0, 0, 0)^T$  respectively, up to an overall phase,

$$U_{element1} = H_1 \otimes H_2 \otimes H_3 \otimes H_4 \otimes H_5 \otimes H_6 \otimes H_7 \otimes H_8 \otimes H_9, \quad (9a)$$

$$U_{element4} = H_1 \otimes H_2 \otimes F_3 \otimes H_4 \otimes F_5 \otimes F_6 \otimes H_7 \otimes H_8 \otimes H_9. \quad (9b)$$

Here we pause to note something rather undesirable. If we consider the measurement of the vertical columns to be a “step” in the iteration (i.e. 1,2,3 or 4,5,6 from Fig.(1, b)), then the MBQC algorithm does not have a fixed, constant iterator. The iterator will in general vary between steps and it is neither sequential nor random as can be seen from the first column of Table(I). The various iterators must be carefully chosen by the oracle at each step. This appears very different from Grover’s iterator. In Grover’s algorithm the iterator is chosen by the oracle at the start of the calculation and then remains a constant regardless of the number of iteration steps. Table (I) shows measurement settings for the 6 qubit measurements that result in each of the 8 possible outputs of the MBQC search. Note the output state is corrected for overall phase, so the tags are not unique. In addition, all measurements presented are presumed to have given the “correct” output (i.e. only entries in section 1 of  $|\Psi_{out}\rangle$  are populated), thus negating the need for trivial feed forward corrections typical in MBQC.

TABLE I: Measurement patterns & resulting output (ignoring phase factors) for the ( $n = 3$ )  $2^n = 8$  MBQC search on a  $3 \times 3$  grid of qubits.

Measurement Sequence ( $\mathcal{I}_1 ; \mathcal{I}_2$ )	Search Element i	Output State $ \Psi_{out,i}\rangle$
(0,0,0 ; 0,0,0)	1	(1,0,0,0,0,0,0,0)
(0,0, $\pi$ ; 0, $\pi$ ,0)	2	(0,1,0,0,0,0,0,0)
(0,0,0 ; 0,0, $\pi$ )	3	(0,0,1,0,0,0,0,0)
(0,0, $\pi$ ; 0, $\pi$ , $\pi$ )	4	(0,0,0,1,0,0,0,0)
(0,0, $\pi$ ; 0,0,0)	5	(0,0,0,0,1,0,0,0)
(0,0,0 ; 0, $\pi$ ,0)	6	(0,0,0,0,0,1,0,0)
(0,0, $\pi$ ; 0,0, $\pi$ )	7	(0,0,0,0,0,0,1,0)
(0,0,0 ; 0, $\pi$ , $\pi$ )	8	(0,0,0,0,0,0,0,1)

## B. The MBQC search algorithm with a single constant iterator for any outcome

We have found a way of modifying the MBQC search algorithm such that the iterator is a constant not only between steps, but also constant for *any* desired output. We begin by noting that in the  $n = 3$  case there is one example in which the iterator is always a constant, namely  $U_{element1}$ . In Eq.(9a) (see first row of Table I) we can see that the two steps of the iterator are the same for  $U_{element1}$ , i.e.  $H_1 \otimes H_2 \otimes H_3$  and  $H_4 \otimes H_5 \otimes H_6$ . To determine the effect of the tagging operation we compare two different  $U$  matrices, for instance  $U_{element4}$  and  $U_{element1}$ . We performed this comparison by the simplest expedient operation of subtraction,

$$\Delta U = U_{element1} - U_{element4} = H^{\otimes 9} - H_1 \otimes H_2 \otimes F_3 \otimes H_4 \otimes F_5 \otimes F_6 \otimes H_7 \otimes H_8 \otimes H_9. \quad (10)$$

Here we find an interesting form for the matrix  $\Delta U$ . Many of the columns are identically zero while the remaining columns in  $\Delta U$  have an absolute value twice that in any given element of  $U_i$ . This implies that the two differently tagged matrices are identical except for the sign of the elements in several vertical columns. This implies that,

$$U_{element1} - U_{element4} T_4 = H^{\otimes 9} - (H_1 \otimes H_2 \otimes F_3 \otimes H_4 \otimes F_5 \otimes F_6 \otimes H_7 \otimes H_8 \otimes H_9) T_8 = 0, \quad (11a)$$

$$T_4 = I_1 \otimes I_2 \otimes (-Z_3) \otimes I_4 \otimes (-Z_5) \otimes (-Z_6) \otimes I_7 \otimes I_8 \otimes I_9, \quad (11b)$$

where in Eq.(11b) we have used the relations  $FF^\dagger = I$  and  $F^\dagger H = i(-Z)$ . The  $i$  in this relation leads to some of the tags being imaginary such as  $T_4$ . Note that this  $i$  is simply an overall phase and without loss of generality can be factored out at any convenient point. For clarity we drop the  $i$  from the definition of  $T_i$  giving all real tags. In the above,  $T_4$  is real a diagonal (tagging) matrix with elements  $\pm 1$ . The 1’s correspond to the columns that are already identical and thus subtract to 0, while the -1 correspond to the remaining columns.  $U_{element1}$  is identical to the product  $U_{element4} T_4$ . Since  $T_i$  is an involution  $T_i T_i = I$ , we can also write  $U_{element1} T_4 = U_{element4}$ . However, the matrices  $T_i$  do not commute with  $U_j$  ( $[T_i, U_j] \neq 0$ ). A matrix  $T_i$  can be found for any matrix  $U_i$ , or equivalently for

any measurement pattern, i.e.  $(s_1, s_2, s_3 ; s_4, s_5, s_6)$ . The  $T_1$  matrix corresponding to the  $U_{element1}$  transformation is equal to the identity matrix  $T_1 = I$ .

As an illustrative example, we provide the explicit form of the  $T$  matrices in the case of the  $(n = 2)$   $2^n = 4$  element search on the box cluster state in Fig. 1a (note: for the  $(n = 3)$   $2^n = 8$  element search, the  $T$  matrices are of size  $2^{n^2} \times 2^{n^2} = 512 \times 512$ ). The diagonal tagging matrices  $T_i$  for  $i \in \{1, 2, 3, 4\}$  corresponds to measurement iterators  $\mathcal{I}_i = (s_1, s_2)$  acting on qubits 1 and 2 given by:  $\mathcal{I}_1 = (0, 0) \rightarrow T_1 = I_1 \otimes I_2 \otimes I_3 \otimes I_4$ ,  $\mathcal{I}_2 = (\pi, 0) \rightarrow T_2 = (-Z_1) \otimes I_2 \otimes I_3 \otimes I_4$ ,  $\mathcal{I}_3 = (0, \pi) \rightarrow T_3 = I_1 \otimes (-Z_2) \otimes I_3 \otimes I_4$ ,  $\mathcal{I}_4 = (\pi, \pi) \rightarrow T_4 = (-Z_1) \otimes (-Z_2) \otimes I_3 \otimes I_4$ , where, for example, the  $T_3$  matrix is given explicitly by  $T_3 = \text{diag}[-1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1]$  and  $T_4 = \text{diag}[-1, -1, -1, -1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1]$ . The placement of the negative signs on the diagonal of  $T_i$  correspond to the  $2^{N-n}$  blocks of size  $2^n$  discussed in the Section (III) and Appendix (A). In this  $n = 2$  example, there are  $2^{4-2} = 4$  sections of size  $2^2 = 4$ .

The difference between  $T_3$  and the other tagging operations  $T_i$  is the location of the negative blocks on the diagonal of the  $T_i$  matrices that are associated with a matrix  $-Z_i$  for each measurement angle  $s_i = \pi$  on qubit  $i$ . Such an oracle is well behaved as defined by Bennet *et. al* [16]. Note that each tagging matrix  $T_i$  is, by construction, a tensor product of  $n^2$   $2 \times 2$  qubit matrices, and therefore a local, non-entangling operation. This differs from Grover's algorithm which can and in general will change the entanglement.

We can now rewrite the fundamental quantum search equation Eq.(5) as,

$$|\Psi_{out,i}\rangle = H^{\otimes n^2} T_i |\Psi_{in}\rangle, \quad (12)$$

where  $T_i$  is the tagging operation chosen by the oracle for the desired output  $i \in \{1, \dots, 2^n\}$ .

Interestingly, the tagging matrix  $T_i$  can be applied either to the right or to the left in Eq.(12). In other words, we can apply the tag to the measurement operation (as typically done in MBQC) thus changing the iterators between iterations, or more interestingly, we can apply the tag to the initial input cluster state resource, prior to the measurement process. Applying the tag to the cluster state leaves the measurement sequence (pattern), or equivalently the MBQC search iterator, constant for any oracle chosen tag, regardless of the size  $N$  of the system, or the desired output of the search algorithm. The constant iterator is simply  $H^{\otimes n}$  acting on each column of qubits in turn.

We now show the effect of acting with the tagging matrix  $T_i$  on the input state. Combining Eq.(1) and Eq.(2) for the  $(n = 2)$  4 element search we have,

$$|\Psi_{in}\rangle = |\Psi_{box}\rangle = \prod_{i,j=nn} CZ_{i,j} |\psi\rangle = CZ_{1,2} CZ_{1,3} CZ_{2,4} CZ_{3,4} \prod_{i=1}^N |+_i\rangle. \quad (13)$$

Applying the  $T_i$  matrix we have,

$$T_i |\Psi_{in}\rangle = T_i |\Psi_{box}\rangle = T_i CZ_{1,2} CZ_{1,3} CZ_{2,4} CZ_{3,4} \prod_{i=1}^N |+_i\rangle. \quad (14)$$

$T_i$  and  $CZ_{i,j}$  are diagonal matrices, and hence commute we have,

$$T_i |\Psi_{in}\rangle = T_i |\Psi_{box}\rangle = CZ_{1,2} CZ_{1,3} CZ_{2,4} CZ_{3,4} T_i \prod_{i=1}^N |+_i\rangle. \quad (15)$$

Thus, we can say that the shape of the cluster state (it's pairwise connectivity), contained within the list of CZ gates, remains the same. The  $T_i$  operation only changes the state of the constituent qubits independently of the connectivity of the cluster state. We find that the effect of the  $T_i$  operation is to change several of signs of the initial cluster state from  $|+\rangle$  states to  $|-\rangle$  states. There is a one to one correlation for a given measurement between those qubits, which in the standard implementation would be measured in the  $B(\pi)$  basis, and the qubits which are rotated from  $|+\rangle$  to  $|-\rangle$ .

As a further example, in the case of the  $(n = 3)$  9 qubit MBQC search the measurement sequence  $(0, 0, \pi ; 0, \pi, \pi)$  on qubits  $(1, 2, 3 ; 4, 5, 6)$  corresponding to the fourth (of eight possible) output state  $(0, 0, 0, 1, 0, 0, 0, 0)^T$  (see Table I). From Eq.(1), the effective initial state for the search algorithm can be written as,

$$T_4 |\psi_{init}\rangle = T_4 \prod_{i=1}^N |+_i\rangle = |+_1\rangle \otimes |+_2\rangle \otimes |-_3\rangle \otimes |+_4\rangle \otimes |-_5\rangle \otimes |-_6\rangle \otimes |+_7\rangle \otimes |+_8\rangle \otimes |+_9\rangle. \quad (16)$$

Note the similarity of Eq. (16) to Eq.(9b) with  $|+_i\rangle \rightarrow H_i$  and  $|-_i\rangle \rightarrow F_i$ . These associations appears to hold true for any system size  $N$  and for any measurement pattern  $(s_1, s_2, s_3 ; s_4, s_5, s_6)$ . That is, any qubit measured in the  $B(\pi)$

basis may equivalently be measured in the standard  $B(0) = |\pm\rangle$  basis if the corresponding qubit in the initial cluster state is flipped from  $|+\rangle \rightarrow |-\rangle$  with an overall phase which is dependent on the total number of  $s_i = \pi$ .

For the ( $n = 3$ )  $N = n^2 = 9$  qubit case, we explicitly display the form sequence of operations leading to MBQC search with a constant iterator. In the following we illustrate the specific example of tagging the fourth output state (i.e. again the  $U_{element4}$  measurement pattern)

$$|\Psi_{out,4}\rangle = U_{element4}|\Psi_{in}\rangle = (H_1 \otimes H_2 \otimes F_3) \otimes (H_4 \otimes F_5 \otimes F_6) \otimes (H_7 \otimes H_8 \otimes H_9) \prod_{i,j=nn} CZ_{i,j} \prod_{i=1}^N |+i\rangle \quad (17a)$$

$$= (H_1 \otimes H_2 \otimes H_3) \otimes (H_4 \otimes H_5 \otimes H_6) \otimes (H_7 \otimes H_8 \otimes H_9) T_4 \prod_{i,j=nn} CZ_{i,j} \prod_{i=1}^N |+i\rangle \quad (17b)$$

$$= (H^{\otimes n})^{\otimes n} \prod_{i,j=nn} CZ_{i,j} T_4 \prod_{i=1}^N |+i\rangle \quad (17c)$$

$$= \prod_{k=1}^n (H^{\otimes n})_k \prod_{i,j=nn} CZ_{i,j} |+1\rangle \otimes |+2\rangle \otimes |+3\rangle \otimes |+4\rangle \otimes |+5\rangle \otimes |+6\rangle \otimes |+7\rangle \otimes |+8\rangle \otimes |+9\rangle. \quad (17d)$$

Thus we have redefined the MBQC search in terms of a single tagging operation  $T_i$  which acts on a constant input state  $\prod_{i=1}^N |+i\rangle$  followed by entangling operations  $CZ_{i,j}$  and a constant iterator defined as  $H^{\otimes n}$  acting on each of the  $n$  columns in turn (from left to right). In fully general terms the equations above become

$$|\Psi_{out,m}\rangle = \prod_{k=1}^n (H^{\otimes n})_k \prod_{i,j=nn} CZ_{i,j} T_m \prod_{l=1}^N |+l\rangle, \quad (18)$$

where  $m$  is the desired output state in the decimal representation found by the tagging operation  $T_m$ ,  $N = n^2$  is the total number of qubits  $n$  is the number of iterations as well as the size of one side of the square cluster-state and  $nn$  is the set of nearest neighbor in that cluster-state. In comparison, Grover's algorithm is quite different. Grover's oracle must repeatedly apply a unique tag to the state at each step so that the inversion about the mean operation can be applied correctly. Without this repeated tagging the algorithm would fail to converge to a unique answer. The MBQC algorithm can be performed with one, and only one, initial tagging operation to converge to a unique output state.

## VI. IS THERE AMPLITUDE AMPLIFICATION IN THE MBQC SEARCH ALGORITHM?

We have now reformulated the MBQC search algorithm such that it utilizes a constant iterator,  $H^{\otimes n}$  applied to  $n$  qubit vertical columns, regardless of the required output state. Grover's iterator  $G = U_{inv} U_f$  consists of two unitary operators: the (externally supplied) tagging operator  $U_f$  chosen by the oracle, and the inversion about the mean operation, which were discussed above in section (IV). The inversion about the mean operator  $U_{inv}$  has entries  $D_{ij} = 2/N_G$  for  $i \neq j$  and  $(U_{inv})_{ii} = -1 + 2/N_G$ . This can be rewritten as  $U_{inv} = 2P' - I$  where  $P' = |\psi_{init}\rangle\langle\psi_{init}|$  is the projection matrix on to the unbiased state  $|\psi_{init}\rangle = \frac{1}{\sqrt{N_G}} \sum_x |x\rangle$  with  $P'_{ij} = 1/N_G$  for all  $i, j$ . The amplitude amplification takes advantage of the sign difference between targeted solution states after the tagging operation, and the re-adjustment of the amplitudes affected by the inversion about the mean operator. The Grover search algorithm requires repeated applications of the tagging operation  $U_f$  because  $U_{inv}$  is Hermitian ( $U_{inv} U_{inv} = I$ ), i.e.  $G^2 = I$  if  $U_f \rightarrow I$  [1].

In contrast, the MBQC search algorithm does not apply any amplitude amplification, nor does it repeatedly tag the state. As shown previously, the MBQC search oracle  $T$  only needs to tag the state once at the beginning of the algorithm. Recall that the MBQC iterator  $U$  described above implicitly contains projective measurements  $P$  similar to Eq. (6). The MBQC iterator is not an invertible operator for any size search algorithm. However, we can treat the MBQC search algorithm effectively implementing a unitary iterator on logical qubits, since the projective measurements performed on the columns of qubits in the  $n \times n$  qubit grid leave the measured qubits in well defined states and the remaining logical qubits in a pure state.

Brassard *et. al* generalized Grover's algorithm to a wider class of unitary amplitude amplification search problems [1, 17]. The general amplitude amplification iterator is defined by [17–19]

$$Q = A U_0 A^{-1} U_f, \quad (19)$$

$U_f$  is the phase flip tagging operation discussed previously,  $U_0$  is the unitary operator that flips the sign of the computational basis state  $|0\rangle^{\otimes n}$  while leaving all other computational basis states unchanged. The operator  $A$  is the generalized unitary oracle constrained only by the requirement that  $A|0\rangle^{\otimes n} = |\psi_{init}\rangle$ . Since the MBQC algorithm inherently requires projective measurements to proceed it can not implement amplitude amplification because it has no equivalent analogue to the inverse  $A^{-1}$ . We have confirmed this numerically by performing incomplete MBQC searches, i.e. we performed less iterations than the algorithm required for successful search completion, and examined the output. We found that the MBQC algorithm, which starts in the entangled cluster state  $|CS\rangle = \prod_{i,j=nn} CZ_{i,j}|\psi_{init}\rangle$ , does not alter the magnitude of any basis state amplitude, between the first and penultimate iteration ( $k = 1, \dots, n-2$ ). Thus no amplitude amplification takes place.

Instead of amplitude amplification the MBQC search appears to be performing a binary search operation. This empirical statement is backed by the computational time required to implement the MBQC algorithm. We find that the MBQC search algorithm requires exactly  $(n-1)$  iterations to find a solution with unit probability. Each MBQC iteration involves a simultaneous measurement of  $n$  qubits, for a total of  $n(n-1)$  single-qubit measurements. If we include the final measurement to read out solution contained in the last column of qubits, the MBQC algorithm has applied  $n$  measurement iterations (measuring a total of  $n^2$  qubits) to effect a search on  $2^n$  elements.

Note that any other smaller number of iterations (i.e.  $k \leq n-1$ ) will not yield a valid search solution, since the system remains in a state unitarily equivalent to the initial cluster state  $|CS\rangle$ . This result has been numerically

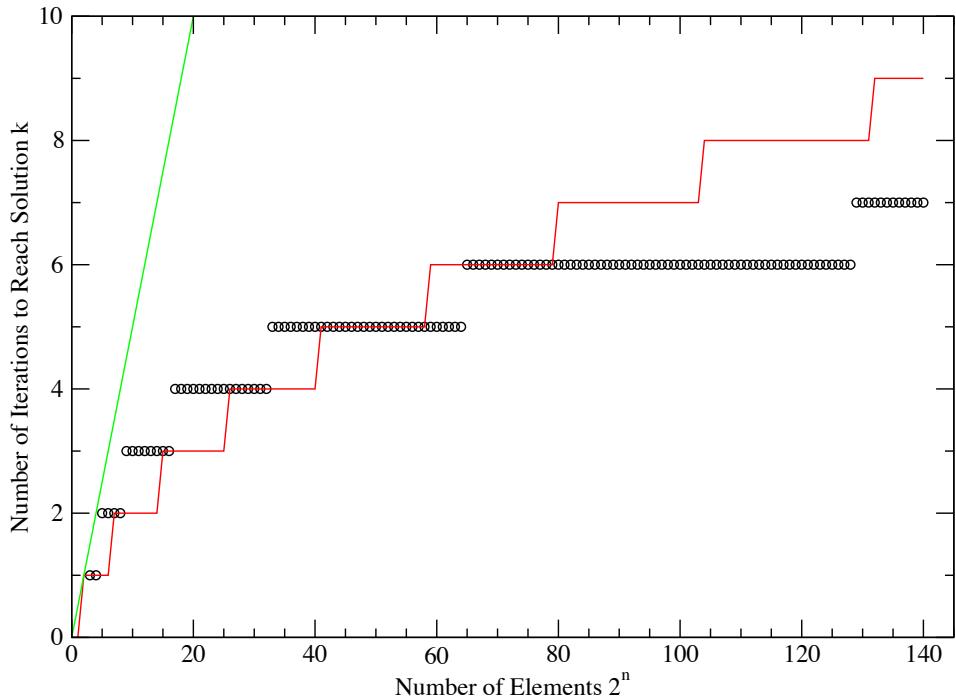


FIG. 2: **Iteration count vs number of searched elements**, for classical and quantum unstructured search algorithms. The number of elements in an unsorted vector and the number of iterations required for three search algorithms. The classical  $2^n/2$  search (straight green line), the standard Grover's algorithm estimate of  $\lfloor (\pi/4)\sqrt{2^n} \rfloor$  (red "stair-step" line) and the MBQC search algorithms  $\log_2 2^n$  (black circles). (Colors Online). Note that the first point at which the MBQC algorithm becomes faster is 59 elements, which requires a 6x6 qubit square grid. The last point at which Grover's is as fast as the MBQC search is an 80 element search on a 7x7 qubit square grid.

tested for searches of  $2^n$  elements with  $n = 2, \dots, 9$ . The ( $n = 9$ )  $2^9 = 512$  element MBQC search requires 81 total qubits, 72 single qubit measurements and up to 72 post rotations. Such a large cluster state is currently physically un-realizable, but our numerical results show that such a cluster state takes only  $n-1 = 8$  applications of the MBQC iterator compared to  $(\pi/4)\sqrt{2^n} \approx 17$  applications of Grover's iterator. This result is reflected in Fig. (2) where the abscissa has been truncated to maintain clarity for the smaller element sizes which are currently achievable.

This suggests that the MBQC search is not amplitude amplification, rather a 1-to- $2^n$  mapping, and that the oracle is simply selecting a given path. As each single-qubit measurement in the MBQC search has only 2 outcomes, the MBQC search algorithm can be thought of as a binary tree structure. The oracle directs the search through the binary tree (from root to leaf) based on the oracle's application of the tagging operation. Each individual measurement in either the 0 or  $\pi$  rotated basis (i.e.  $B(0)$  or  $B(\pi)$  from Eq.(8)) selects a branch of the binary tree to move down. As

stated above, some measurements are not necessary if one ignores or corrects for overall phase effects. In fact all  $2^n$  outcomes for the MBQC search algorithm can be found with a minimum of exactly  $n$  variable measurements. This is suggestive of the binary structure, since the total number of possible unique outcomes is  $2^n$ .

## VII. ENTANGLEMENT DYNAMICS

### A. Relative entropy of entanglement

As another quantitative means of comparison between the MBQC algorithm and Grover's algorithm, we calculated the relative entropy of entanglement for the logical three qubit and four qubit searches. For Grover's algorithm this is relatively straightforward. We need only record the system state vector  $|\psi\rangle$  at any given stage of the calculation, form the density matrix  $\rho = |\psi\rangle\langle\psi|$ , and calculate the relative entropy relative to the closest separable density matrix  $\sigma$ . The MBQC calculation is slightly more involved as one must identify the current  $n$  qubit logical pure state within the  $n \times n$  qubit cluster state during the computations, after each projective measurement.

We calculate the three qubit and four qubit relative entropy of entanglement for each step of the respective searches numerically via a numerical optimization routine [20]. The relative entropy of entanglement is based on the Von Neumann entropy,

$$S(\rho) = -\text{Tr}(\rho \ln \rho). \quad (20)$$

We use the relative entropy of entanglement because we desire a measure of the distinguishability of our two search states compared to the set of all separable states. The relative entropy of entanglement is defined as [20, 21],

$$E_q(\rho) = \min_{\sigma \in S_q} -\text{Tr}(\rho \ln \rho - \rho \ln \sigma), \quad (21a)$$

$$= \min_{\sigma \in S_q} \langle \psi | -\ln \sigma | \psi \rangle \quad (21b)$$

where  $\sigma$  is an element of the set of all  $q$ -separable states  $S_q$ , namely states that are separable for  $q$  partitions of the  $n$ -qubit system, and  $\rho$  is the density matrix representing the logical state of the system [22]. In Eq. (21b) we have used the fact that  $\rho = |\psi\rangle\langle\psi|$  is a pure state density matrix. For  $E_n$ ,  $\sigma$  is  $n$ -qubit separable.  $E_n$  quantifies the amount of entanglement of any kind contained in the global system, i.e. it is non-vanishing even for states showing entanglement just between two subsystems [21, 23]. For  $E_2$ , which quantifies genuine multipartite entanglement [21, 23], the minimization of  $\sigma$  takes place over all possible bipartite partitions.

For the  $n = 3$  qubit case, there are only three possibilities, (i)  $E_1 = 0$  (since  $\sigma$  is a pure 3-qubit state), (ii)  $E_2$  in which  $\sigma$  is bipartite separable and (ii)  $E_3$  in which  $\sigma$  is 3-qubit separable: We define the set of separable states  $\sigma$  numerically as,

$$\sigma_{(q=2)}^{(n=3)} = \sum_{i=1}^{N_G^2} p_i (|\phi_{i,1,2}\rangle |\phi_{i,3}\rangle) (\langle \phi_{i,1,2}| \langle \phi_{i,3}|), \quad (22a)$$

$$\sigma_{(q=3)}^{(n=3)} = \sum_{i=1}^{N_G^2} p_i (|\phi_{1,i}\rangle |\phi_{2,i}\rangle |\phi_{3,i}\rangle) (\langle \phi_{1,i}| \langle \phi_{2,i}| \langle \phi_{3,i}|), \quad (22b)$$

where  $p_i$  is a probability with  $\sum p_i = 1$ , and  $N_G = 2^n$  is the number of distinct search elements. In Eq. (22a)  $|\phi_{i,1,2}\rangle$  represents a general two-qubit state on qubits 1 and 2 and  $|\phi_{i,3}\rangle$  represents a general single-qubit state on qubit 3. The other possible grouping of qubit labels, i.e.  $\{(1,3), 2\}$  and  $\{(2,3), 1\}$  differ from  $\sigma_{(q=2)}^{(n=3)}$  by local unitary swap operations of the form  $U_1 \otimes U_2 \otimes U_3$ , and hence are effectively searched over in the minimization procedure due to the symmetry of the quantum search problem. In Eq. (22b)  $|\phi_{j,i}\rangle$  is a general single-qubit state on qubit  $j \in \{1, 2, 3\}$ . Note that for  $n$  qubits the set of all density matrices is dimension  $N_G \times N_G$ , and therefore any density matrix can be written as a linear combination of  $N_G^2$  basis matrices (e.g.  $(M_{i,j})_{a,b} = \delta_{i,a} \delta_{j,b}$ , for  $i, j, a, b \in \{1, \dots, N_G\}$ ).

For the  $n = 4$  case,  $E_3$  is a straightforward generalization of Eq. (22a) with an extra single-qubit partition on qubit 4, and  $E_4$  is a straightforward generalization of Eq. (22b) now involving four single-qubits. The relative entropy  $E_2$  now involves two types of bipartite partitions of the form

$$\sigma_{(q=2)}^{(n=4)} = \sum_{i=1}^{N_G^2} p_i (|\phi_{i,1,2}\rangle |\phi_{i,3,4}\rangle) (\langle \phi_{i,1,2}| \langle \phi_{i,3,4}|), \quad (23a)$$

$$\sigma_{(q=3)}'^{(n=4)} = \sum_{i=1}^{N_G^2} p_i (|\phi_{1,2,3,i}\rangle |\phi_{4,i}\rangle) (\langle \phi_{1,2,3,i}| \langle \phi_{4,i}|). \quad (23b)$$

Both forms of separable density matrices in Eq. (23a) and Eq. (23b) must be searched over in the minimization procedure. We calculate the relative entropy of entanglement value by numerically minimizing over the set of separable states  $\sigma$  and take advantage of the fact that the relative entropy is convex thus a local search results in a global minimum. We have preformed this calculation for each step of the two search algorithms for both the  $n=3$  qubit 8 element search and the  $n=4$  qubit 16 element search.

## B. The pure state system density matrix

The use of a pure state system density matrix  $\rho$  in the standard Grover's algorithm this is straightforward as we are dealing with a pure state vector of size  $2^n$  at each iteration. As stated above for the MBQC search the use of a pure state  $\rho$  is slightly more complicated and requires some interpretation regarding the logical  $n$  qubit state in the space of the larger cluster state. From Raussendorf, Browne and Briegel [24] we learn that the MBQC algorithm can be broken into a series of projective measurements and entangling operations. This can be seen in the following relation,

$$(P_2 D_2)(P_1 D_1) = (P_2 P_1)(D_2 D_1). \quad (24)$$

Here  $\{P_i\}$  is a set of  $n$  qubit projective measurements, thus removing a column of qubits, and  $\{D_i\}$  is a set of entangling operations [24] acting on the logical state and adding new qubits to the cluster state. The key point to note is that the  $D_i$ 's commute with the  $P_k$ 's. This fact allows for the piecewise construction and computation on cluster states. In other words, one can implement the MBQC search algorithm (ignoring whether or not it is the same as Grover's algorithm) while never requiring the presence more than  $2n$  qubits at the same time; the additional qubits can be entangled "on the back end" and operated upon at a later time. Stated slightly differently, the feed forward operations in the MBQC implementation never effect more than nearest neighbor qubits. We used this feature in our calculation of larger cluster state such as the 81 qubit cluster state discussed in section (VI).

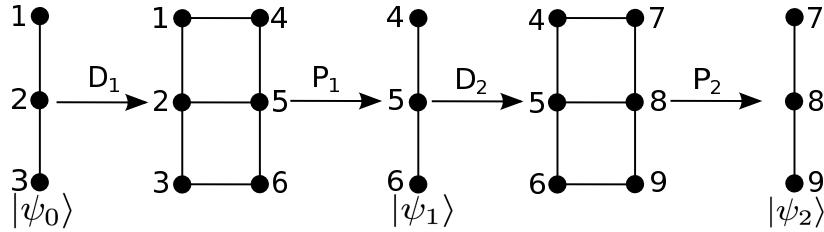


FIG. 3: **Schematic of piecewise cluster state implementation.** Cluster state map and effective pure state vector in MBQC algorithm (see text for description).

In the ( $n = 3$ )  $2^3$  element MBQC search implemented on a  $3 \times 3$  grid of qubits illustrated in Fig. (3), the first three qubits 1,2,3 are initialized into the  $|+\rangle$  state yielding the unbiased state  $|\psi_{init}\rangle$  (the same state utilized in the 3-qubit Grover search algorithm at iteration  $k = 0$ ). Subsequently, the qubits are  $CZ$ -entangled with each other, yielding the pure state  $|\psi_0\rangle = CZ_{1,2}CZ_{2,3}|\psi_{init}\rangle \equiv |C3\rangle$ .  $|\psi_0\rangle$  is the initial system pure state of the MBQC search algorithm at iteration  $k = 0$ , and is the three qubit linear cluster state  $|C3\rangle$ . We expect this state to be highly entangled, which can be seen in Fig. (4a), where we plot the relative entropy  $E_2$  for both Grover's algorithm (red) and the MBQC algorithm (black) for 3 qubits, versus the iteration number  $k$ . The MBQC search iterator then begins its first iteration ( $k = 1$ ) by entangling  $|\psi_0\rangle$  with qubits 4,5,6, which are also begin in the  $|+\rangle$  state, and are themselves  $CZ$ -entangled with each other, i.e.  $D_1$ . This creates a  $2 \times n$  cluster state, similar to the butterfly state [25]. The qubits 1,2,3 are then projectively measured, i.e.  $P_1$ . This leaves a three qubit state on 4,5,6 which we denote as the pure state  $|\psi_1\rangle$ . (Recall that without loss of generality we can consider qubits  $i = (1, 2, 3)$  to be projected into their respective  $|+_{s,i}\rangle$  state, with any required Pauli corrections taking place in post-processing). This is now the three qubit system pure state after one iteration ( $k = 1$ ) of the algorithm. In the next iteration ( $k = 2$ ) the state  $|\psi_1\rangle$  is entangled with three new qubits 7,8,9, identically to the previous step. For clarity we call this unitary operation  $D_2$ . Qubits 4,5,6 are then measured,  $P_2$ , and the logical pure state  $|\psi_2\rangle$  is contained in qubits 7,8,9. We note that the projective operations  $P_i$  are simply the three qubit projective iterators discussed in section (V), and the  $D_i$  are the piecewise construction, i.e. entanglement, of the cluster state illustrated in Fig. (3). The qubits 7,8,9 need not exist when qubits 1,2,3 are measured since the quantum information is entirely stored in 4,5,6. The consequence is that no more than  $2n$  qubits are ever needed at any given time in the algorithm. This has particular relevance to cluster states constructed from "flying" qubits. The significance is that no more than  $2n$  photons need ever be present simultaneously. This constitutes a significant advantage for the photon-based MBQC architectures, especially since

the creation of large photonic cluster states is currently limited by the difficulty inherent in producing larger number of on-demand single photons.

The above piecewise constructive procedure also yields a clear sequence of three and four qubit logical *pure* states,  $\{|\psi_0\rangle, |\psi_1\rangle, |\psi_2\rangle, \dots\}$  that we can compare directly with corresponding pure states produced in the standard Grover's algorithm. We have carried out this comparison using the relative entropy of entanglement  $E_2$  as defined above. The results are shown in Fig. (4). Here the number of iterations has been increased beyond the ideal number of steps for successful search completion for the 8 element and the 16 element searches in order to reveal more behavior of the algorithms.

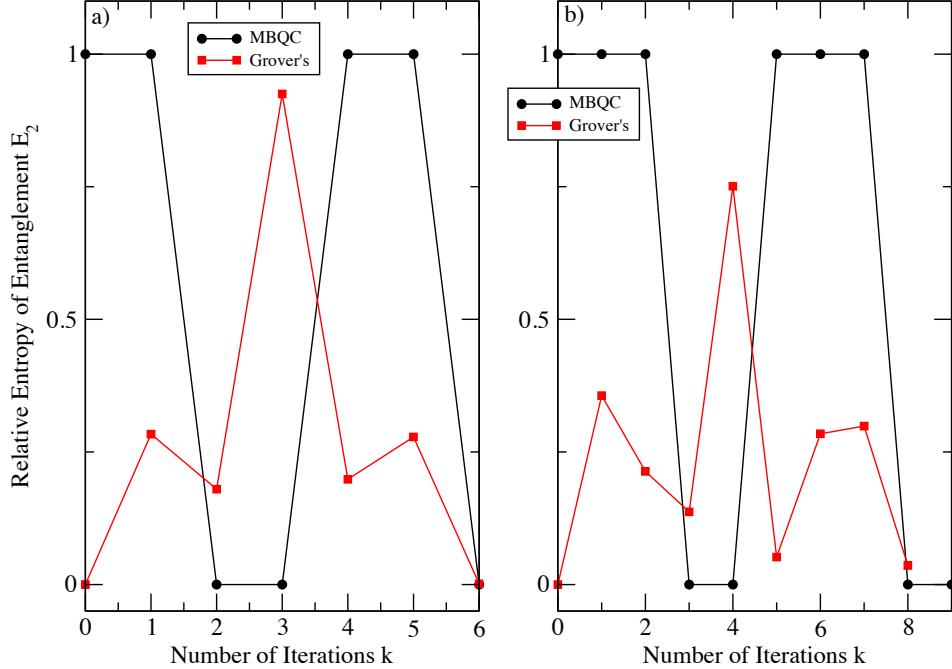


FIG. 4: **Relative Entropy of entanglement  $E_2$  vs. iteration number  $k$**  for (a) the three and (b) the four qubit logical states of both quantum search algorithms. Red squares, the relative entropy of entanglement of Grover's Algorithm. Black circles, the relative entropy of entanglement of the MBQC search algorithm. Note that while the red curve in figure (a) appears symmetric, it is not, nor are the states with similar  $E_2$  identical.

### C. Relative entropy analysis for the Grover search algorithm

We expect the entanglement of the standard Grover's algorithm to be  $E_2 = 0$  for the input state, an  $n$ -qubit separable product state, and then increase to a maximum after  $\lfloor(\pi/4)\sqrt{2^n}\rfloor$  steps (where  $\lfloor x \rfloor$  is the nearest integer less or equal to  $x$ ) before returning close to zero again on the solution state (approximately an  $n$ -qubit separable product state). This was shown by M. Rossi *et. al* [21]. This is confirmed by our results for the first 2 steps (Grover iterations of  $U[G]$ ) in Fig. (4a) and the first 3 steps of Fig. (4b) (red). Note the graph for the standard Grover's algorithm doesn't return to  $E_2 = 0$  at the ideal terminal iteration ( $k = 2$  for the 8 element search, and  $k = 3$  for the 16 element search) where the solution is achieved, because the standard algorithm is not perfect at this step, and we are not performing any projective measurements on the output states. Iterating past the solution step, the spike in the entanglement at step  $k = 3$  in Fig. (4a) and step  $k = 4$  in Fig. (4b) is due to the imperfect nature of Grover's algorithm. It is interesting that the most entangled state is never utilized in Grover's original algorithm. G. L. Long proposed a modified Grover's algorithm that is almost perfect (i.e terminates on a solution that is almost exactly an  $n$  qubit separable state) for larger size systems [8], that should eliminate this spike.

The general behavior of Grover's algorithm can be understood from its periodic behavior with produces the state  $|\psi_k\rangle = U_G^k |\psi_{init}\rangle$  after the  $k$ -th application of the Grover iterate  $U_G$ . As is well known [17–19], the unbiased state can be decomposed into the two-state basis  $|\psi_{init}\rangle = \sin \theta |\psi_{good}\rangle + \cos \theta |\psi_{bad}\rangle$ . Here,  $|\psi_{good}\rangle = |x_0\rangle$  is the normalized solution state (chosen to be  $|0\rangle_1 |1\rangle_2 |0\rangle_3$  for the graph in Fig. (4a)), and  $|\psi_{bad}\rangle = \frac{1}{\sqrt{N_G-1}} \sum_{x \neq x_0} |x\rangle$ , with  $\sin \theta = \frac{1}{\sqrt{N_G}}$  is the superposition of all the non-solution states. After  $k$  iterations of  $U_G$ , the resulting state is given by  $|\psi_k\rangle = U_G^k |\psi_{init}\rangle$

$= \sin(2k+1)\theta|\psi_{good}\rangle + \cos(2k+1)\theta|\psi_{bad}\rangle$ , with the optimal number of steps for a solution given by the condition that  $(2k+1)\theta = \pi/2$  yielding the above estimate  $k_{Grover} \sim \lfloor (\pi/4)\sqrt{2^n} \rfloor$ . In general, a Grover search solution will be obtained under the condition  $(2k+1)\theta = (2m+1)\pi/2$ , for any integer  $m$ . In Fig. (4a) the second occurrence of  $E_2 \approx 0$  in the graph for Grover's algorithm occurs for  $k = 6$  and  $k = 8$  for the 3-qubit and 4-qubit search respectively, yielding estimates of the integers  $m$  (utilizing  $\theta \approx \{1/\sqrt{8}, 1/\sqrt{16}\}$ ) of  $m \approx \{0.987, 0.902\}$ , both roughly close to  $m = 1$ . The peak (spike)  $E_2 \approx 1$  occurs at  $k = 3$  and  $k = 4$  for the 3-qubit and 4-qubit search respectively, with  $(2k+1)\theta = (0.716\pi, 0.788\pi)$ . These values are both roughly  $3\pi/4$  associated with the maximally entangled state  $|\psi_{E_2=1}\rangle = (|\psi_{good}\rangle - |\psi_{bad}\rangle)/\sqrt{2}$ , which has the same magnitude for all amplitudes (as in the unbiased state  $|\psi_{init}\rangle$ ), but with an important relative minus sign between the solution and non-solution states. Finally, the secondary maxima of  $E_2$  for Grover's algorithm for the 3-qubit case occur for iterations  $k = 1$  and  $k = 5$ . At these steps, all computational basis state amplitudes have the same sign, and all have equal magnitude except for the solution state component whose magnitude is significantly larger (but whose square is not close enough to unity for the whole state to be approximately  $|x_0\rangle$ ). Similar behavior occurs for the secondary  $E_2$  maxima in the 4-qubit case.

#### D. Relative entropy analysis for the MBQC search algorithm

We expect the MBQC state, which starts in the 3-qubit case in a highly entangled state  $|\psi_0\rangle = |C3\rangle \equiv CZ_{1,2}CZ_{2,3}|+\rangle_1|+\rangle_2|+\rangle_3$ , to decrease in entanglement as the projective measurements are made. Here we'll write the state vector at the  $k$ -th iteration  $|\psi_k\rangle$  in the  $|\pm\rangle$  representation so that  $|\psi_0\rangle = [1, 0, 1, 0, 0, 1, 0, -1]/\sqrt{8}$  with the first component corresponding to  $|+\rangle_1|+\rangle_2|+\rangle_3$  and the last component corresponding to  $|-\rangle_1|-\rangle_2|-\rangle_3$ . In relation to Grover's algorithm which begins in unbiased state  $|\psi_0\rangle = |\psi_{init}\rangle$ , the MBQC puts all the entanglement in the initial state at iteration  $k = 0$  with  $E_2 \equiv 1$  as shown the black curves in Fig. (4). For the 3-qubit case shown in Fig. (4a) (with tag  $T_3$  selected, i.e. a constant iterator of  $(\pi, \pi, \pi)$ , see the last row of Table IV in Appendix C), the relative entropy of the state  $|\psi_1\rangle = i[-1, 1, 1, 1, -1, 1, 1, 1]/\sqrt{8}$  is unchanged after the first MBQC iteration and remains at  $E_2 = 1$ . At the second ( $n-1=2$ ) iteration the state  $|\psi_2\rangle$  is transformed into the solution state  $|x_0\rangle = [0, 0, 1, 0, 0, 0, 0, 0]/\sqrt{8} = |+\rangle_1|-\rangle_2|+\rangle_3 = H^{\otimes 3}|0_1\rangle|1_2\rangle|0_3\rangle$  with  $E_2 \equiv 0$ . Continuing to iterate past the successfully completed search, the state after the third iteration  $|\psi_3\rangle = i[-1, 1, -1, 1, 1, -1, 1, -1]/\sqrt{8} = F^{\otimes 3}|\psi_2\rangle$  again has  $E_2 \equiv 0$ . At the fourth iteration, we calculate  $|\psi_4\rangle = [-1, 0, 1, 0, 0, -1, 0, -1]/\sqrt{8}$  with  $E_2 \equiv 1$ , very similar in structure to the initial cluster state  $|C3\rangle$ . Additionally, the two iterations  $k = 4, 5$  produce states similar in structure and relative entropy  $E_2 \equiv 1$  as in iterations  $k = 0, 1$ . In fact, the states at  $k = \{0, 1, 4, 5\}$  are local unitary equivalent to each other, i.e. differ on by local unitary transformations of the form  $U_1 \otimes U_2 \otimes U_3$ . Finally, iteration  $k = 6$  returns again to a relative entropy  $E_2 \equiv 0$  with output state  $|\psi_6\rangle = [0, 0, 0, 0, 0, 0, 1]/\sqrt{8} = |-\rangle_1|-\rangle_2|-\rangle_3 = H^{\otimes 3}|1_1\rangle|1_2\rangle|1_3\rangle$ . The state  $|\psi_6\rangle$  is the computational basis solution state that would have arisen at iteration  $k = 2$  if one had initially implemented the tagging operation  $T_8$  (see the third row of Table IV in Appendix C). The states at  $k = \{2, 3, 6\}$  are local unitary equivalent to each other. Qualitatively similar behavior is exhibited in the 4-qubit case as shown in Fig. (4b).

The above cyclic behavior of the “solution sets” of the MQBC algorithm is evident from Table IV for the 3-qubit case and Table V for the 4-qubit case in Appendix C. In these tables we repeat the entangling/rotation/projection operations many steps beyond iteration  $k = n-1$  where the solution associated with the chosen tag first appears, i.e. the state  $|x = i\rangle$ , in the decimal representation, associated with tag  $T_i$  for  $i \in \{1, \dots, 2^n - 1\}$ . For a given tag  $T_i$ , the solution either (i) stays the same (modulo overall phase) or (ii) periodically cycles between two states (modulo factors of  $\pm 1$  and/or  $\pm i$ ) every  $n+1$  iterations. Note that the “orbits” for different tags may cross on these solutions. That is, a given solution state  $|x\rangle$  (e.g.  $x = (2, 4, 5, 7)$  in Table (IV)) may appear in the “orbits” of more than one tag due to the differences in the measurement bases involved in the MBQC iterators for each chosen tag.

The MBQC states in Fig. (4a) and Fig. (4b) with  $E_2 = 1$  can be understood as follows. For the 3-qubit case at  $k = 0$  the initial state is the cluster state  $|C3\rangle = (|0_1\rangle|C2_{2,3}\rangle + |1_1\rangle|C2'_{2,3}\rangle)/\sqrt{2}$  with  $|C2_{2,3}\rangle = (|+\rangle_2|0_3\rangle + |-\rangle_2|1_3\rangle)/\sqrt{2}$  and  $|C2'_{2,3}\rangle = (|-\rangle_2|0_3\rangle + |+\rangle_2|1_3\rangle)/\sqrt{2}$ . A representative bi-separable state yielding  $E_2 = 1$  is given by  $\sigma_{(q=2)}^{(n=3)} = 1/2(|0_1, C2_{2,3}\rangle\langle 0_1, C2_{2,3}| + |1_1, C2'_{2,3}\rangle\langle 1_1, C2'_{2,3}|)$ , which is the sum of the projectors associated with each of the two terms in  $|C3\rangle$ . This state is bi-separable in the qubit partition  $\{1, (2, 3)\}$ , while exhibiting maximal entanglement on the  $(2, 3)$  sub-partitions embodied in  $|C2_{2,3}\rangle$  and  $|C2'_{2,3}\rangle$ . All other bi-separable states with  $E_2 = 1$  are local unitary equivalent to  $\sigma_{(q=2)}^{(n=3)}$ . The first measurement does not change the value of  $E_2$  so that the state at  $k = 1$  is again local unitary equivalent to  $\sigma_{(q=2)}^{(n=3)}$ . Similar behavior occurs in the 4-qubit case. At  $k = 0$  the cluster state is given by  $|C4\rangle = (|0_1\rangle|+\rangle_2|C2_{3,4}\rangle + |1_1\rangle|-\rangle_2|C2''_{3,4}\rangle)/\sqrt{2}$  with the same form as before and  $|C2''_{3,4}\rangle = (|+\rangle_3|0_4\rangle - |-\rangle_3|1_4\rangle)/\sqrt{2}$ . A representative bi-separable state yielding  $E_2 = 1$  is given by  $\sigma_{(q=2)}^{(n=4)} = 1/2(|0_1, +_2, C2_{3,4}\rangle\langle 0_1, +_2, C2_{3,4}| + |1_1, -_2, C2''_{3,4}\rangle\langle 1_1, -_2, C2''_{3,4}|)$ . Again, all other bi-separable states with  $E_2 = 1$  are

local unitary equivalent to  $\sigma_{(q=2)}^{(n=4)}$ , as are the states at  $k = 1$  and  $k = 2$ .

### VIII. REMAINING QUESTIONS

We have examined a number of aspects of the MBQC search algorithm, however, several questions remain unanswered. How can a search be preformed that has more than one tagged output in the MBQC algorithm? How does this affect the speed of the algorithm? Can the idea of critical damping be applied to such multi-element MBQC algorithm? Can the database search algorithm on “itemized lists” encoded in tensor products states as described at the end of Section IV A [15] be implemented in a MBQC algorithm? Can the MBQC algorithm be implemented on a blind cluster state [26, 27] with similar speed? These questions will be explored in future work.

### IX. CONCLUSION

Grover’s Algorithm and the MBQC search algorithm (that has been called Grover’s algorithm in the literature [10]) are qualitatively similar in form but significantly different in structure and operation. Both search algorithms appear to be oracular search methods, mapping a chosen tagging operation, performed by the oracle, with a corresponding unique search solution. However, the behavior of the oracle in each search algorithm is quite different; in Grover’s algorithm the tagging is an entangling operation and is performed at each iteration, while in the MBQC search algorithm the tagging operation is non-entangling and can be performed on the unbiased initial state, before the application of the entangling  $CZ$  gates used to create the initial cluster state. The iterators of the two search algorithm are also structurally very different. In Grover’s search algorithm, the iterator preforms amplitude amplification, while in the MBQC search algorithm the iterator appears to be a selection a of single “route” or “path” in a  $1$ -to- $2^n$  mapping of an initial state the set of all possible solution states. This is most readily exhibited through the entanglement dynamics presented in this work of the  $n$ -qubit logical state at each iteration. We have shown explicitly that the iterators in the MBQC search algorithm do not perform amplitude amplification; the projective measurements toggles between states of the same entanglement class, either similar to the entangled initial state, or similar to the  $n$ -qubit separable solution state. Significantly, the speed of the two search algorithm are very different, with the MBQC search being the faster of the two. That is, for a  $2^n$  element search, Grover’s search algorithm requires  $\mathcal{O}(2^{n/2})$  applications of the Grover iteration acting on  $n$ -qubits. For the MBQC algorithm acting on  $n^2$  qubits, only  $n - 1$ , i.e.  $\mathcal{O}(n)$  applications of the MBQC iterator are required, acting on  $n$ -qubits at each iteration. From the above analysis and observations we draw the conclusion that MBQC search algorithm is not Grover’s algorithm, and should be studied in its own right.

### Acknowledgments

This work was supported in part by the National Research Council (NRC) Research Associateship program at the AFRL Rome Research Site, Information Directorate and the Air Force Office of Scientific Research (AFOSR). We are gratefully for useful discussions with M. Fanto. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of AFRL.

### References

---

- [1] L. Grover, Proc. 28th annual ACM Sym. on Theory of Comp. (STOC) pp. 212–219 (1996).
- [2] P. W. Shor, SIAM J. Sci. Statist. Comp. **26**, 1484 (1997).
- [3] P. Kok and *et al.*, Rev. Mod. Phys. **79**, 135 (2007).
- [4] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, arXiv:quant-ph/0001106v1 (2000).
- [5] H. J. Briegel and *et. al.*, Nature Physics **5**, 19 (2009).
- [6] H. J. Briegel and R. Raussendorf, Phys. Rev. Lett. **86**, 910 (2001).
- [7] C. Zalka, Phys. Rev. A **60**, 2746 (1999).
- [8] G. L. Long, Phys. Rev. A **64**, 022307 (2001).

- [9] D. P. DiVincenzo, *Fortschr. Phys.* **48**, 771 (2000).
- [10] P. Walther and *et. al*, *Nature* **434**, 169 (2005).
- [11] C. Schmid, N. Kiesel, W. Wieczorek, and H. Weinfurter, *New J. Phys.* **9**, 236 (2007).
- [12] M. A. Nielson and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge Univ. Press, N.Y., 2000).
- [13] D. E. Brown and H. J. Briegel, “One-way Quantum Computation” in *Lectures on Quantum Information* (eds. D. Bruß and G. Leuchs, Wiley-VCH, N.Y., 2007).
- [14] N. S. Yanofsky and M. A. Mannucci, *Quantum Computing for Computer Scientists* (Cambridge Univ. Press, N.Y., 2008).
- [15] P. M. Alsing and N. McDonald, *Proc. SPIE* **8057**, 80570R (2011).
- [16] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, *SIAM J. on Computing* **26**, 1510 (1997).
- [17] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, *Quant. Comp. and Quantum Info.: A Millennium Volume* **305** (2002).
- [18] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, arXiv:quant-ph/9805082v1 (1998).
- [19] P. Kaye, R. Laflamme, and M. Mosca, *An Introduction to Quantum Computing* (Oxford Univ. Press, N.Y., 2007).
- [20] K. Brádler, M. Wilde, S. Vinanampathy, and D. B. Uskov, *Phys. Rev. A* **82**, 062310 (2010).
- [21] M. Rossi, D. Bruß, and C. Macchiavello, arXiv:1205.3000v1 (2012).
- [22] V. Vedral, M. B. Plenio, M. A. Rippin, and P. Knight, *Phys. Rev. Lett.* **78**, 12 (1997).
- [23] M. Blasone, F. Dell’Anno, S. D. Siena, and F. Illuminati, *Phys. Rev. A* **77**, 062304 (2008).
- [24] R. Raussendorf, D. E. Browne, and H. J. Briegel, *Phys. Rev. A* **68**, 022312 (2003).
- [25] S. Y. Ma and *et. al*, *Optics Comm.* **283**, 497 (2010).
- [26] A. Broadbent, J. F. Fitzsimons, and E. Kashefi, arXiv:0807.4154v3 (2009).
- [27] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, *Science* **335**, 303 (2012).

### Appendix A: Pauli corrections and the projective measurement $\mathbf{P}$

In Section (III) we state that the output matrix  $|\psi_{out}\rangle = U|\psi_0\rangle$  is a vector of size  $2^N$  (see Eq. (5)). We then claim that without loss of generality we can consider the logical output state to always be contained within the first  $2^n$  elements of the output vector, and ignore the remaining  $2^{N-n}$  components. This statement follows from the non-deterministic projective nature of quantum mechanical measurements. Whenever we attempt to make a measurement on a superposition of two states (say the  $|+\rangle$  and  $|-\rangle$  states) we can obtain either output. In the MBQC implementation one makes a partially ordered series of single-qubit measurements on the  $|+_s\rangle$  state ( $|+_0\rangle = |+\rangle$  and  $|+_n\rangle = |-\rangle$ ). For each single-qubit measurement there is equal probability that a particular outcome will yield the “incorrect”  $|-_s\rangle$  measurement, ( $|-_0\rangle = |-\rangle$  and  $|-_n\rangle = |+\rangle$ ). This does not present a problem since the incorrect measurement merely adds a trivial phase and Pauli flip  $X$  to the desired target state. Eq. (A1) shows the effect described in the main body for the case of the  $n=2$ , 4-element, search, where measurements will be performed on qubits 1 and 2 as in Fig. 1, with the output state projected on qubit 3 and 4. Equation (A1) show the entire 4-qubit state prior to the projection operation. Here we use the unitary matrix (tagging and iteration) that yields the 4th element  $(0, 0, 0, 1)^T$  on qubits 3 and 4 (analogous to Eq.(9b)).

$$|\psi_{out}\rangle = U_{element4}|\psi_0\rangle = \frac{1}{2} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \hline 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ -1 \\ 0 \\ -1 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} |\psi_{target}\rangle \\ \hline -I \otimes X |\psi_{target}\rangle \\ \hline -X \otimes I |\psi_{target}\rangle \\ \hline -X \otimes X |\psi_{target}\rangle \end{pmatrix}. \quad (A1)$$

Notice that the desired logical output is indeed contained with the first  $2^n$  elements of  $|\psi_{out}\rangle$  (the “first section”). The remaining  $2^{N-n}$  terms contain  $n-1$  copies (sections) of the targeted state  $|\psi_{target}\rangle$  (on qubits 3 and 4), each of which has undergone a unique set of Pauli bit flips and overall phase manipulations corresponding to those single-qubit measurements (on qubit 1 and 2) that were projected into the incorrect states  $|-_s\rangle$ . In effect we are calculating every possible outcome of the  $n(n-1)$  measurements simultaneously, and therefore the state before the projection is an equal superposition of all these possible outcomes. An experiment would produce only one of these four output states with equal probability. Since we know the outcome of the measurements (i.e. we observe the measurement record), we always know which particular qubits were projected into the state  $|-_s\rangle$ . Thus, we can always construct the Pauli bit flip and overall phase operations shown in Eq. (A1), to correct (in post processing) the output such that we end up with exactly  $|\psi_{target}\rangle$  on qubits 3 and 4. For operational simplicity and computational speed, we truncate our matrices  $U$  to the first  $2^n$  rows of the full  $2^N \times 2^N$  unitary matrix, thus containing one copy of the “correct” (i.e. not Pauli flipped or phase shifted) logical output. This is implemented numerically by simple truncation of the matrix  $U$  and justified logically by stating that we can assume, without loss of generality (since we can perform the necessary Pauli corrections in post processing), the measured qubits are always projected into the correct  $|+_s\rangle$  states. This is equivalent to applying the operator  $P = |+_s\rangle\langle +_s|$  (identical to Eq.(6)) on each each measured qubit.

### Appendix B: Measurement setting for the MBQC search

As illustrative examples we provide measurement settings for the MBQC search for the ( $n = 3$ ) 8 and ( $n = 4$ ) 16 element searches. Table (II) shows the 8 element search similar to the main body, but additionally also shows the setting which provide constant iterators. Observe that in both sets of measurements there are only 3 free variable

measurements. Note the the (trivial) overall phase factors which are ignored in this table are not the same between the two measurements settings. This demonstrates part of the  $n$ -fold degeneracy that was mentioned in the section on the tagging operation.

TABLE II: Measurement sequences & resulting output ignoring phase factors for  $2^3$  elements

Constant Iterators ( $\mathcal{I}_1 ; \mathcal{I}_2$ )	Variable Iterators	Element i	Output State $ \Psi_{out,i}\rangle$
(0,0,0 ; 0,0,0)	(0,0,0 ; 0,0,0)	1	(1,0,0,0,0,0,0,0)
(0, $\pi$ , $\pi$ ; 0, $\pi$ , $\pi$ )	(0, $\pi$ ; 0, $\pi$ ,0)	2	(0,1,0,0,0,0,0,0)
( $\pi$ , $\pi$ , $\pi$ ; $\pi$ , $\pi$ , $\pi$ )	(0,0,0 ; 0,0, $\pi$ )	3	(0,0,1,0,0,0,0,0)
( $\pi$ ,0,0 ; $\pi$ ,0,0)	(0,0, $\pi$ ; 0, $\pi$ , $\pi$ )	4	(0,0,0,1,0,0,0,0)
( $\pi$ , $\pi$ ,0 ; $\pi$ , $\pi$ ,0)	(0,0, $\pi$ ; 0,0,0)	5	(0,0,0,0,1,0,0,0)
( $\pi$ ,0, $\pi$ ; $\pi$ ,0, $\pi$ )	(0,0,0 ; 0, $\pi$ ,0)	6	(0,0,0,0,0,1,0,0)
(0,0, $\pi$ ; 0,0, $\pi$ )	(0,0, $\pi$ ; 0,0,0)	7	(0,0,0,0,0,0,1,0)
(0, $\pi$ ,0 ; 0, $\pi$ ,0)	(0,0,0 ; 0, $\pi$ , $\pi$ )	8	(0,0,0,0,0,0,0,1)

Table (III) shows the 16 element search and the constant iterator measurement set. Here we are able to show the phase factors for these measurements.

TABLE III: Measurement sequences & resulting output including phase factors for  $2^4$  elements

Constant Iterators ( $\mathcal{I}_1 ; \mathcal{I}_2 ; \mathcal{I}_3$ )	Element i	Output State $ \Psi_{out,i}\rangle$
(0,0,0,0 ; 0,0,0,0 ; 0,0,0,0)	1	(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
( $\pi$ , $\pi$ , $\pi$ ,0 ; $\pi$ , $\pi$ , $\pi$ ,0 ; $\pi$ , $\pi$ , $\pi$ ,0)	2	(0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
( $\pi$ ,0, $\pi$ , $\pi$ ; $\pi$ ,0, $\pi$ , $\pi$ ; $\pi$ ,0, $\pi$ , $\pi$ )	3	(0,0,-i,0,0,0,0,0,0,0,0,0,0,0,0,0)
(0, $\pi$ ,0, $\pi$ ; 0, $\pi$ ,0, $\pi$ ; 0, $\pi$ ,0, $\pi$ )	4	(0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0)
( $\pi$ , $\pi$ ,0, $\pi$ ; $\pi$ , $\pi$ ,0, $\pi$ ; $\pi$ , $\pi$ ,0, $\pi$ )	5	(0,0,0,0,-i,0,0,0,0,0,0,0,0,0,0,0)
(0,0, $\pi$ , $\pi$ ; 0,0, $\pi$ , $\pi$ ; 0,0, $\pi$ , $\pi$ )	6	(0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0)
(0, $\pi$ , $\pi$ ,0 ; 0, $\pi$ , $\pi$ ,0 ; 0, $\pi$ , $\pi$ ,0)	7	(0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0)
( $\pi$ ,0,0,0 ; $\pi$ ,0,0,0 ; $\pi$ ,0,0,0)	8	(0,0,0,0,0,0,0,-i,0,0,0,0,0,0,0,0)
(0, $\pi$ , $\pi$ , $\pi$ ; 0, $\pi$ , $\pi$ , $\pi$ ; 0, $\pi$ , $\pi$ , $\pi$ )	9	(0,0,0,0,0,0,0,-i,0,0,0,0,0,0,0,0)
( $\pi$ ,0,0, $\pi$ ; $\pi$ ,0,0, $\pi$ ; $\pi$ ,0,0, $\pi$ )	10	(0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0)
( $\pi$ , $\pi$ ,0,0 ; $\pi$ , $\pi$ ,0,0 ; $\pi$ , $\pi$ ,0,0)	11	(0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0)
(0,0, $\pi$ ,0 ; 0,0, $\pi$ ,0 ; 0,0, $\pi$ ,0)	12	(0,0,0,0,0,0,0,0,0,0,-i,0,0,0,0,0)
( $\pi$ ,0, $\pi$ ,0 ; $\pi$ ,0, $\pi$ ,0 ; $\pi$ ,0, $\pi$ ,0)	13	(0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0)
(0, $\pi$ ,0,0 ; 0, $\pi$ ,0,0 ; 0, $\pi$ ,0,0)	14	(0,0,0,0,0,0,0,0,0,0,0,0,0,-i,0,0)
(0,0,0, $\pi$ ; 0,0,0, $\pi$ ; 0,0,0, $\pi$ )	15	(0,0,0,0,0,0,0,0,0,0,0,0,0,0,-i,0)
( $\pi$ , $\pi$ , $\pi$ , $\pi$ ; $\pi$ , $\pi$ , $\pi$ , $\pi$ ; $\pi$ , $\pi$ , $\pi$ , $\pi$ )	16	(0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1)

### Appendix C: Cyclical behavior of the constant iterator MBQC search algorithm

We find that the MBQC search is also cyclical in nature when a constant iterator is applied. For a given tag, Grover's algorithm cycles between the desired target state, an  $n$ -qubit separable state, and a different solution state which approximates an  $n$ -qubit separable state. In contrast, the behavior of the MBQC search is more complicated. For a given tag, the desired target state will be produced at iteration  $k = n - 1$ . Afterwards, for some tags, the desired target state will be produced every  $n + 1$  iterations. However for other tags, the states cycle every  $n + 1$  iterations between the target state and a second distinct valid output state that is exactly an  $n$ -qubit separable state. Note that the desired target state repeats with period  $2(n + 1)$  regardless of the chosen tag. Table (IV) shows the chosen iterator and the resulting tagged element from 1 to  $2^3 = 8$  for the case  $n = 3$ , including overall phases at the output of each cycle. Table (V) shows the corresponding results for the case  $n = 4$ . The output states in Table (IV) are similar to Fig.(4a) at iterations  $k = 2$  and  $k = 6$ , and beyond (similarly for Table (V) and the output states at iterations  $k = 3$  and  $k = 8$  in Fig.(4b)). Thus, each column requires a different size cluster state or a different number of piece wise operations  $P_i$  and  $D_i$  similar to Eq.(24). For the  $n = 3$  case, the iterators  $(0, 0, 0)$  and  $(\pi, 0, \pi)$ , corresponding to the

TABLE IV: The tagged element and the number of iterations at which it is reached for the  $2^3$  element search.

Constant Iterator $\mathcal{I}$	Iteration 2	Iteration 6	Iteration 10	Iteration 14	Iteration 18
(0,0,0)	1	1	1	1	1
(0,0, $\pi$ )	-7	2	7	-2	-7
(0, $\pi$ ,0)	8	-3	8	-3	8
(0, $\pi$ , $\pi$ )	-2	-4	2	4	-2
( $\pi$ ,0,0)	-4	5	4	-5	-4
( $\pi$ ,0, $\pi$ )	6	6	6	6	6
( $\pi$ , $\pi$ ,0)	-5	-7	5	5	-5
( $\pi$ , $\pi$ , $\pi$ )	3	-8	3	-8	3

target states 1 and 6 respectively, give output elements that are always identical. The remaining tags cycle between between two  $n$ -qubit separable target states. The precise cause of this cycling behavior is a topic warranting further investigation.

TABLE V: The tagged element and the number of iterations at which it is reached for the  $2^4$  element search.

Constant Iterators $\mathcal{I}$	Iteration 3	Iteration 8	Iteration 13	Iteration 18	Iteration 23	Iteration 28
(0,0,0,0)	1	1	1	1	1	1
( $\pi$ , $\pi$ , $\pi$ ,0)	-15i	2	15i	-2	-15i	2
( $\pi$ ,0, $\pi$ , $\pi$ )	-12i	3	12i	-3	-12i	3
(0, $\pi$ ,0, $\pi$ )	6	4	6	4	6	4
( $\pi$ , $\pi$ ,0, $\pi$ )	-14i	5	14i	-5	-14i	5
(0,0, $\pi$ , $\pi$ )	-4	6	-4	6	-4	6
(0, $\pi$ , $\pi$ ,0)	-7	7	-7	7	-7	7
( $\pi$ ,0,0,0)	-9i	8	9i	-8	-9i	8
(0, $\pi$ , $\pi$ , $\pi$ )	-8i	9	8i	-9	-8i	9
( $\pi$ ,0,0, $\pi$ )	10	10	10	10	10	10
( $\pi$ , $\pi$ ,0,0)	-13	11	-13	-11	-13	11
(0,0, $\pi$ ,0)	3i	12	-3i	-12	3i	12
( $\pi$ ,0, $\pi$ ,0)	11	13	11	13	11	13
(0, $\pi$ ,0,0)	5i	14	-5i	-14	5i	14
(0,0,0, $\pi$ )	-2i	15	2i	-15	-2i	15
( $\pi$ , $\pi$ , $\pi$ , $\pi$ )	-16	16	-16	16	-16	16